5M

УДК 004.272

Т.В. Михайлова
Донецький національний технічний університет, м. Донецьк, Україна
tanya@r5.dgtu.donetsk.ua

Програмна система для моделювання ієрархічної комп'ютерної пам'яті

Запропоновано імітаційну модель для моделювання протоколів когерентності. Визначено вихідні дані для моделювання ієрархічної комп'ютерної пам'яті. За результатами моделювання визначено кращий протокол для конкретного класу задач, що обчислюються в багатопроцесорному середовищі.

Вступ

Підвищення продуктивності обчислювальних систем безпосередньо пов'язане зі збільшенням швидкодії і ємності пам'яті. Ємність пам'яті найбільш великих обчислювальних систем зросла від 1000 байт до десятків терабайт, а час циклу зменшився з 20 мкс до 10 нс. Швидкодіючі запам'ятовувальні пристрої залишаються більш дорогими, чим повільні. З метою зменшення вартості при тій же продуктивності ефективніше мати ієрархію пам'яті з невеликим за ємністю запам'ятовувальним пристроєм [1], розташованим поруч із процесором, що й мають мінімальний час доступу. Така ієрархія дозволяє узгодити характеристики пам'яті й центрального процесора.

Оскільки швидкість процесорів та кількість ядер, що використовуються на сучасних пристроях, збільшується, актуальність кешу теж збільшується.

Системи для аналізу роботи кешу ще не набули достатньої популярності. Крім того, існуючі зараз системи не дозволяють зазирнути у середину роботи процесора, можна проаналізувати лише загальні відомості про процесор та його кеш.

Сучасні умови потребують систем, які б могли аналізувати ефективність кешу для різних класів задач, що обчислюються на багатопроцесорній або багатоядерній техніці.

Мета роботи полягає в розробці програмної системи, за допомогою якої можна аналізувати ефективність кешу для різних класів задач, що обчислюються на багатопроцесорній або багатоядерній техніці.

Підтримка когерентності кеш-пам'яті

При використанні ієрархічної пам'яті для кожного елемента даних повинна бути забезпечена когерентність (погодженість, однаковість) його копій, у різних рівнях ієрархії. Для багатопроцесорної ЕОМ справа ускладнюється й тим, що одні й ті ж дані можуть вимагатися різними процесорами.

Реалізація механізму когерентності в ОМ із розподіленою пам'яттю вимагає апаратурно-тимчасових витрат. Зменшити тимчасову складову витрат можна за рахунок збільшення апаратурної складової і навпаки. Зменшення тимчасової складової вимагає створення спеціалізованої апаратури реалізації когерентності. Зменшення апаратурної

складової передбачає деякий мінімум апаратних засобів, на яких здійснюється програмна реалізація механізму когерентності. Розглянемо деякі протоколи.

Для забезпечення когерентності пам'яті моделі із роздільними кешами й загальною пам'яттю можна використовувати алгоритм MESI (Modified, Exclusive, Shared, Invalid) [2]. Він являє собою організацію когерентності кеш-пам'яті зі зворотним записом. Цей алгоритм запобігає зайвим передачам даних між кеш-пам'яттю й основною пам'яттю. Так, якщо дані в кеш-пам'яті не змінювалися, то нема чого їх пересилати.

Кожний рядок кеш-пам'яті може перебувати в одному з наступних станів: *M* – рядок модифікований (доступний до читання й запису тільки в цьому ОМ, тому що модифікований командою запису порівняно з рядком основної пам'яті); *E* – рядок монопольно копійований (доступний до читання й запису в цьому ОМ і в основній пам'яті); *S* – рядок багатокопійований або розподілений (доступний до читання й запису в цьому обчислювальному модулі (ОМ), в основній пам'яті й у кеш-пам'яті інших ОМ, у яких утримується її копія); *I* – рядок, неможливий до використання (рядок не доступний ні до читання, ні до запису).

Стан рядка використовується як для визначення процесором ОМ можливості локального, без виходу на шину, доступу до даних у кеш-пам'яті, так і для керування механізмом когерентності.

Для керування режимом роботи механізму підтримки когерентності використовується біт WT, стан 1 якого задає режим наскрізного (write-through) запису, а стан 0 – режим зворотного (write-back) запису в кеш-пам'ять [2].

При виконанні команд читання й запису стан рядка кеш-пам'яті, до якого виконується доступ, визначається MESI-діаграмою [2].

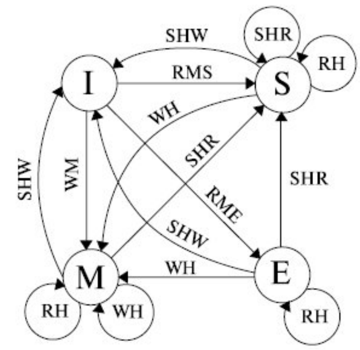


Рисунок 1 – MESI-діаграма забезпечення когерентності кеш-пам'яті

Алгоритм підтримки когерентності пам'яті MOESI дозволяє уникнути необхідності запису змін із кеш-пам'яті в основну для дозволу іншому процесору їх читання завдяки стану *Owned*, в якому процесор передає модифіковані дані безпосередньо в інший процесор.

В стані *Owned* містяться коректні дані. Тільки один із процесорів може мати дану кеш-лінію в стані *Owned*, усі інші процесори можуть мати ці дані тільки в стані *S*. Зміна стану за даним протоколом наведена в [3].

Стан *Owned* надав всі переваги процесору-хазяїнові, але інші процесори в положенні тільки для читання. А якщо кому-небудь із них було необхідно зробити запис у свій рядок зі станом *S*, то доводилося робити її й в оперативну пам'ять, що приводило до ліквідації стану *Owned* в іншого процесора. У багатопотокових середовищах, де різні процесори або процесорні ядра можуть одночасно виконувати різні потоки

того самого процесу, це представляє проблему. Для поліпшення продуктивності в цьому випадку уведений стан *Written* в алгоритмі підтримки когерентності пам'яті MOWESI. При записі в рядок зі станом *S* він змінювався на *Written*, інші процесори сповіщалися про факт і зобов'язувалися ліквідувати або обновити свої застарілі копії, після чого стан рядка змінювався на *M* або *Owned*.

Реалізація та тестування моделі

Загальний алгоритм роботи кожного кешу обчислювального модуля в моделі наведено на рис. 2. Квант модельного часу дорівнює такту процесора. Вихідні дані для моделі наступні: *r* – вірогідність читання; 1-*r* – вірогідність запису; *p* – вірогідність влучення до кешу; 1-*p* вірогідність невлучення до кешу; вірогідності знаходження даного в станах *M*, *O*, *W*, *E*, *S*, *I* (в залежності від класу задач, що обчислюються), ємність кешу. Реалізовано три алгоритми підтримки когерентності пам'яті: MESI, MOESI, MOWESI.

```
Если (1-p) то
        //данного в кэш нет
        Определение статуса данного.
        Пересылка данного из ОП или другого кэш.
        Обработка данного (чтение/запись) по протоколу.
          Изменение статуса данного.
Иначе
        //данное в кэш
        Определение статуса данного.
        Если (r) то
            Чтение данного по соотв.протоколу.
            Изменение статуса данного.
        Иначе
            Запись данного в соответствии с протоколом.
            Изменение статуса данного.
```

Рисунок 2 – Загальний алгоритм роботи кеш

Як тестовий процесор був обраний Intel Core i5 [4] як один з найпопулярніших на цей час.

На рис. 3 наведено середній час обробки заявки (в тактах). В цьому тесті однакова вірогідність находження даного в різних станах (для спрощення), що не зовсім відповідає дійсності, тому середній час великий. З рисунку видно, що в цьому випадку кращим протоколом є MOWESI, завдяки зменшенню часу обміну з оперативною пам'яттю. Тестування моделі ускладнюється багатьма параметрами вихідних даних.

Висновки

Проаналізоване функціонування протоколів кешів для багатопроцесорних систем. Зроблено висновок, що чим складніший є процесор, тим більш складного протоколу когерентності він потребує. Розроблена і протестована модель для різних протоклів когерентності.

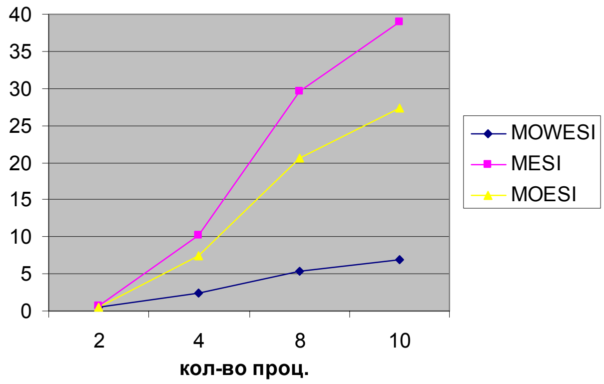


Рисунок 3 – Середній час обробки заявки

В подальшому планується розширити кількість протоколів, що підтримуються програмою, а також кількість процесорів, що можуть бути промодельовані. Є можливість доопрацювання системи для моделювання поведінки кешу будь-якої системи, що містить кеш.

Подальші дослідження направлені на аналіз ефективності кешу відповідно класу задач, що обчислюються на паралельній техніці.

Література

1. Столингс У. Структурная организация и архитектура компьютерных систем / Столингс У. – М. : Вильямс, 2002. – 893 с.
2. Цилькер Б.Я. Организация ЭВМ и систем / Б.Я. Цилькер, С.А. Орлов. – М. : ПИТЕР, 2004. – 668 с.
3. MOWESI : Материал из Википедии – свободной энциклопедии : Версия 25099283, сохранённая в 03:10 UTC 4 мая 2010 // Википедия, свободная энциклопедия [Электронный ресурс]. – Сан-Франциско : Фонд Викимедиа, 2010. – Режим доступа : http://en.wikipedia.org/wiki/MOWESI_protocol.
4. Обзор и тестирование процессоров Intel Core i5-750 и Core i7-870 [Электронный ресурс]. – Режим доступа : http://www.ferra.ru/online/system/91487/

Т.В. Михайлова

Программная система для моделирования иерархической памяти

Предложена имитационная модель для моделирования протоколов когерентности. Определены исходные данные для моделирования иерархической компьютерной памяти. По результатам моделирования выбран лучший протокол для конкретного класса обрабатываемых задач.

T.V. Mikhaylova

Programm System for Modeling Hierarchical Memory

The simulation model is suggested for modeling protocol of data storage. Raw data for modeling hierarchical computer memory is determined. The best protocol is chosen on the result of modeling certain type of the processed problems.

Стаття надійшла до редакції 14.07.2010.